How to Git(Hub)?

What is Git and GitHub?

Git: Tool that tracks changes in your code: Version control. We can use it to update and maintain code on our local machine and directly upload it from our code editor/terminal.

GitHub: Is where you host all your repositories/projects. It can be published and reused by other group members and others. On GitHub a directory with project files is called a repository.

Useful commands when working with the command line/terminal:

Changing the directory

The “cd” command is used to change the directory. It can be used to directly access a target directory or do it step-by-step.

Examples

Lets assume the folder structure /Users/user/desktop/my_folder/my_file

If we are in the desktop directory, we can:

- Access my_folder by typing cd my_folder
- Access my_file by typing cd my_folder/my_file
- Access my_file by typing cd ~/my_file
- Access user by typing cd ..
- Access Users by typing cd ../..
-

Clearing the terminal output

The “clear” command will remove all visible output from the terminal. !IT WILL NOT UNDO ANYTHING YOU HAVE DONE!

Within a .git directory, we can additionally:

- Use “git clone *GitHub link*” to clone a repository from GitHub
- Use “git add *filename.abc* to start tracking a specific file
- Use “git add .” to start tracking all files within the current directory
- Use “git commit -m “my commit message”” to commit our changes and get the ready to be uploaded.
- Use “git push origin main” to upload the committed changes from our local machine to the main branch of the online repository.
- Use “git pull” to download changes made to the online repository to our local machine, if the respective repository has been previously cloned.

Working with Git and GitHub?

GitHub – Setup:

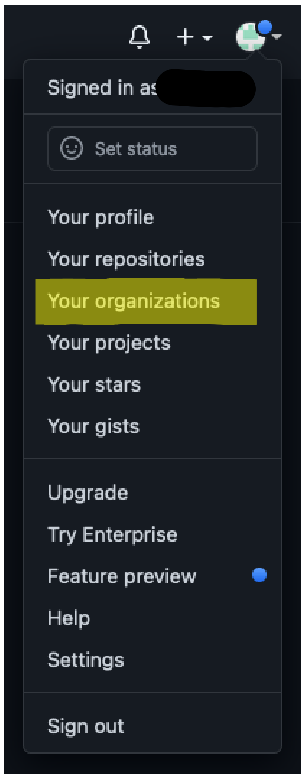
- Create a GitHub account.
- After you log in you will be on the main page or your profile page.

- Ask Karl to add you to the Gademann Lab GitHub Organization

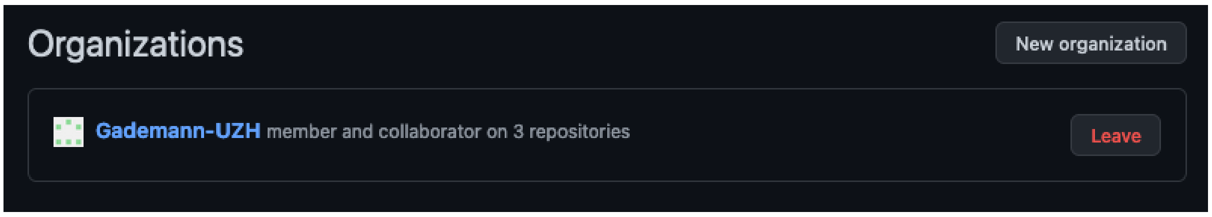
If you plan on using GitHub only as “Cloud Storage” you are done with set up. This means you will have to replace files on GitHub if you update them locally on your machine. In this case, go to **GitHub as “Cloud Storage”**. If you want to be able to update files via the command line from your machine you will also need to set up Git.

GitHub as “Cloud Storage”

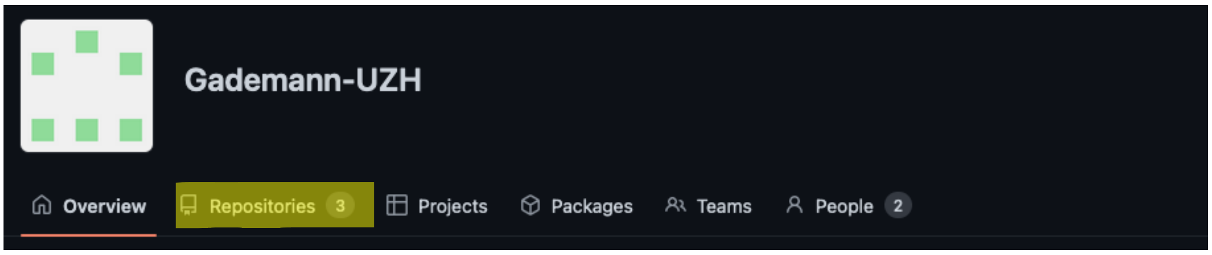
- To use the Gademann group GitHub to store your code-based projects navigate to the group “organization”. You can go there by clicking on your profile picture in the top right corner to open the drop-down menu and click on “Your organizations”



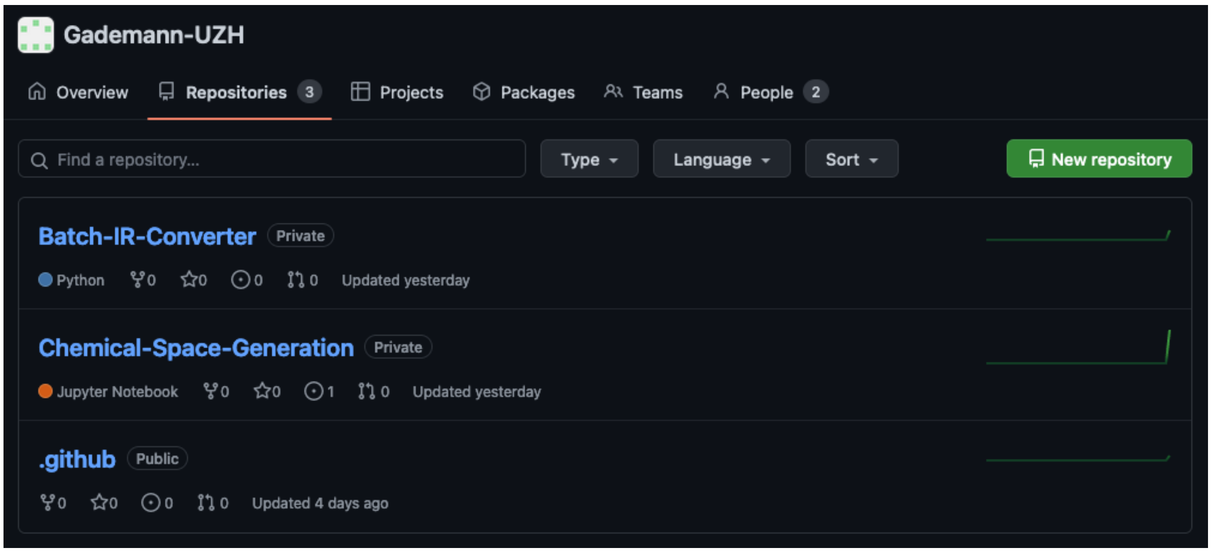
- On this page you will find a list with all organizations you are part of. Click on the Gademann-UZH to go to the group GitHub main page.



- At the top of the group page click on “Repositories”. This will show you an overview of all repositories in the group GitHub.



- To make a new repository click on “New repository”. On the following page give the repository a meaningful name, tick the box to add a README file, and then “Create repository”.

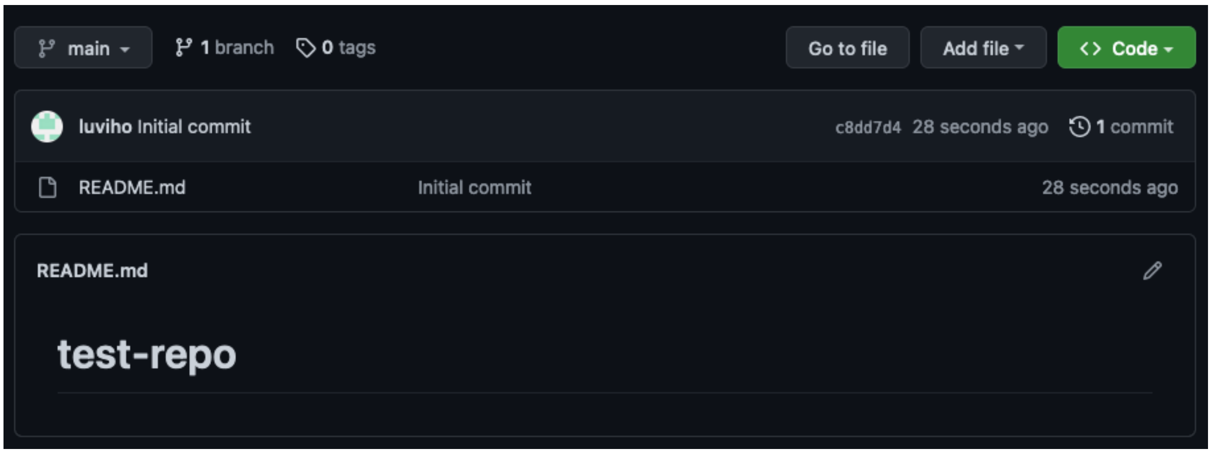


Batch-IR-Converter Private
Python 0 0 0 0 Updated yesterday

Chemical-Space-Generation Private
Jupyter Notebook 0 0 1 0 Updated yesterday

.github Public
0 0 0 0 Updated 4 days ago

- The resulting repository will have a list of all files on top and the automatically generated README.md file displayed below.



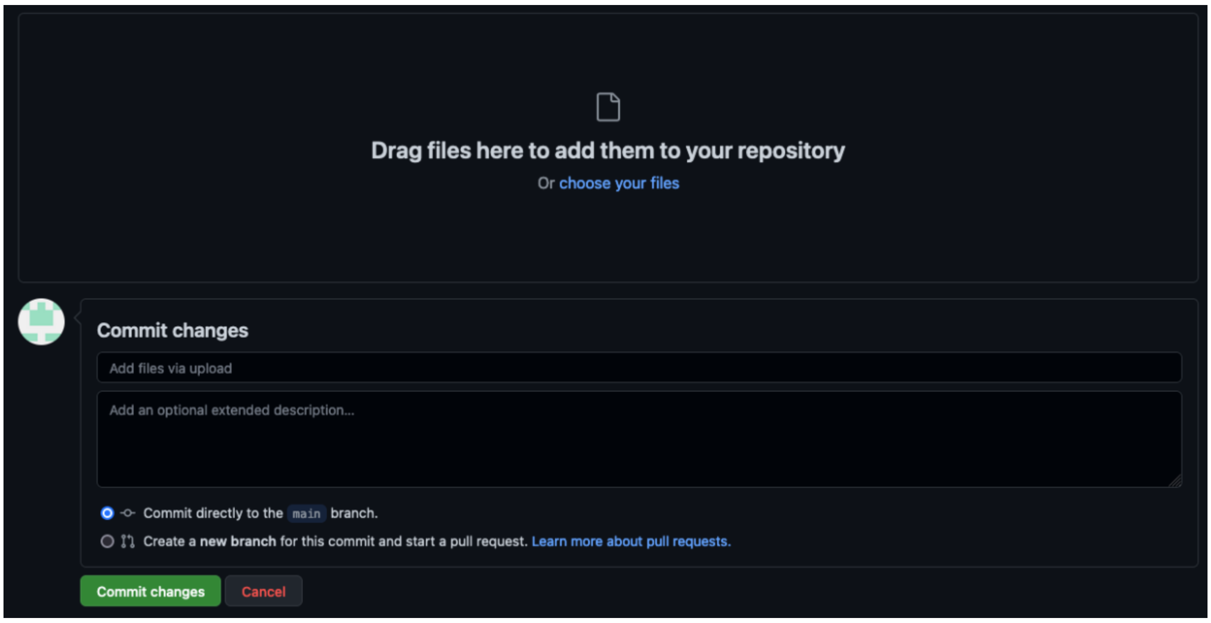
luviho Initial commit c8dd7d4 28 seconds ago 1 commit

README.md Initial commit 28 seconds ago

README.md

test-repo

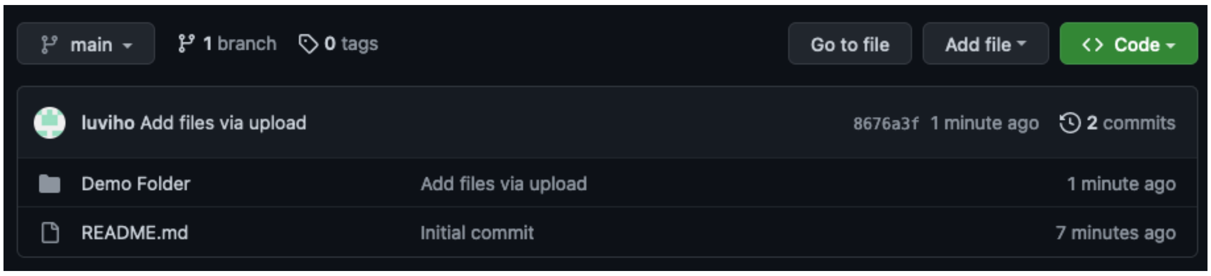
- You can now upload your file by clicking on “Add file” and the on “Upload files”. This will open a new window in which you can choose the files you want to upload. To complete this step scroll to the bottom of that page and click on “Commit changes”



Drag files here to add them to your repository
Or choose your files

Commit changes
Add files via upload
Add an optional extended description...
Commit directly to the main branch.
Create a new branch for this commit and start a pull request. Learn more about pull requests.

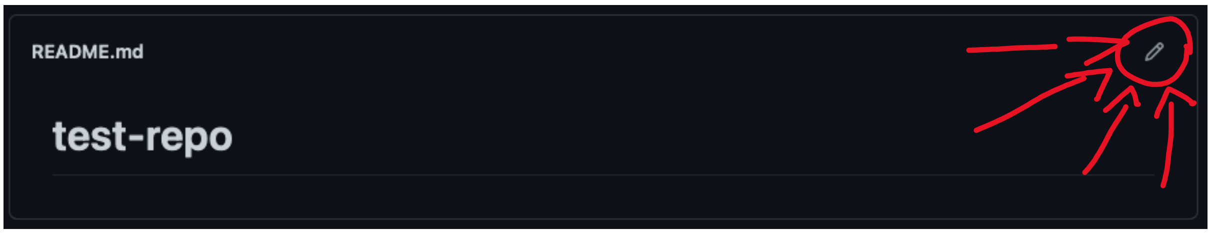
- After your files have been uploaded you will automatically return to the repository, where you can now see the new files in the list of files.



luviho Add files via upload 8676a3f 1 minute ago 2 commits

Demo Folder	Add files via upload	1 minute ago
README.md	Initial commit	7 minutes ago

- ATTENTION! Now that you have uploaded your code/project, make sure to write a proper documentation on how to use your code, program, or script. This is classically done in README files. To edit the one that was automatically generated for you, simply click on the edit symbol in the top right corner. When you are done writing the documentation or you want to save your progress, you can commit your changes at the bottom of the page.



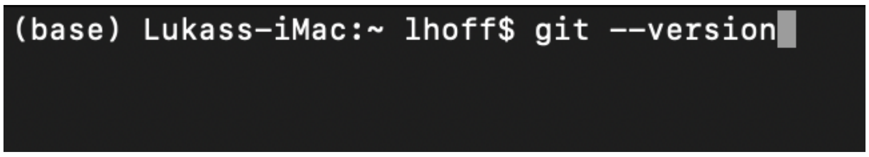
- Congrats! You just created a repository with your project.

Git – Installation:

On Mac OS git should be already installed. You can test this by opening the Terminal and typing the git --version command. If Git is installed this is what it should roughly look like:

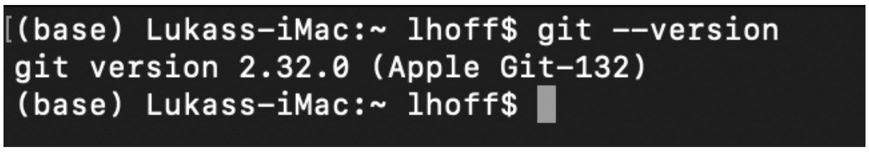
- Open the Terminal:

- Type git --version:



```
(base) Lukass-iMac:~ lhoff$ git --version
```

- Terminal response should state Git version if installed:



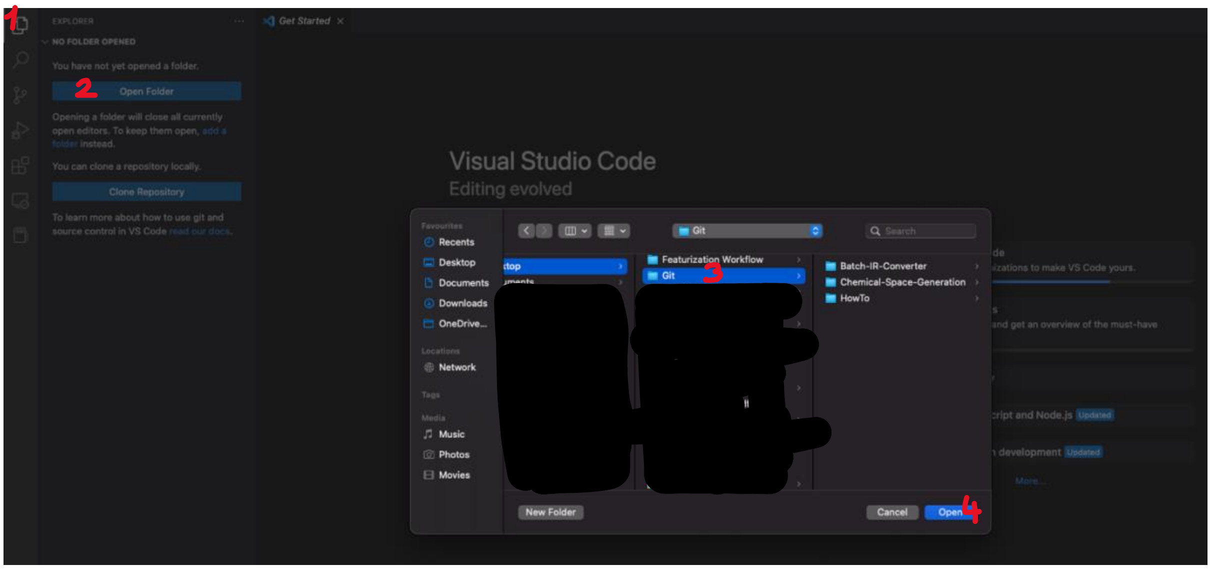
```
(base) Lukass-iMac:~ lhoff$ git --version
git version 2.32.0 (Apple Git-132)
(base) Lukass-iMac:~ lhoff$
```

If Git is not installed or you are using a different OS then install Git. For this you can follow the instructions here: https://www.atlassian.com/git/tutorials/install-git.

Code editor – Setup

To make use of Git you will need to install a code editor, e.g. Visual Studio Code, or any other you like. The following demonstration uses Visual Studio Code.

Make a designated folder for your Git repositories, i.e. ./Documents/Git, or whatever you prefer. Once you created the directory you can open this folder in Visual Studio Code.



You have not yet opened a folder.

Opening a folder will close all currently open editors. To keep them open, add a folder instead.

You can clone a repository locally.

To learn more about how to use git and source control in VS Code read our docs.

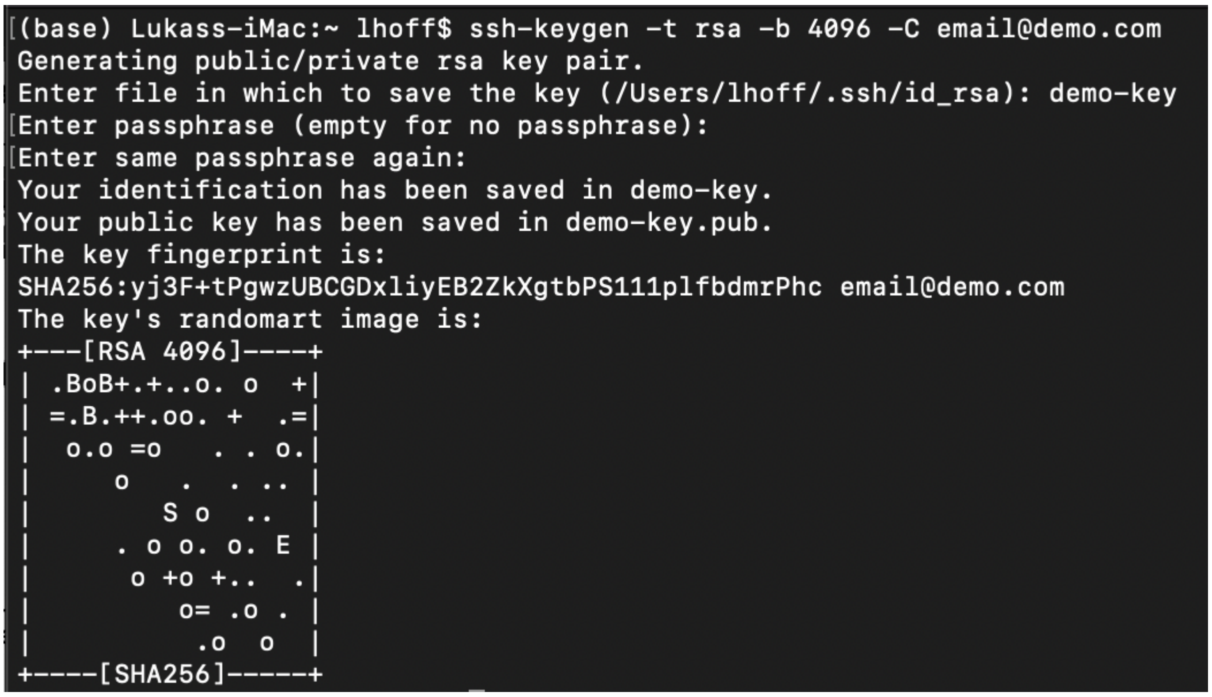
[redacted]

You are now ready to set up Git.

Git – Setup

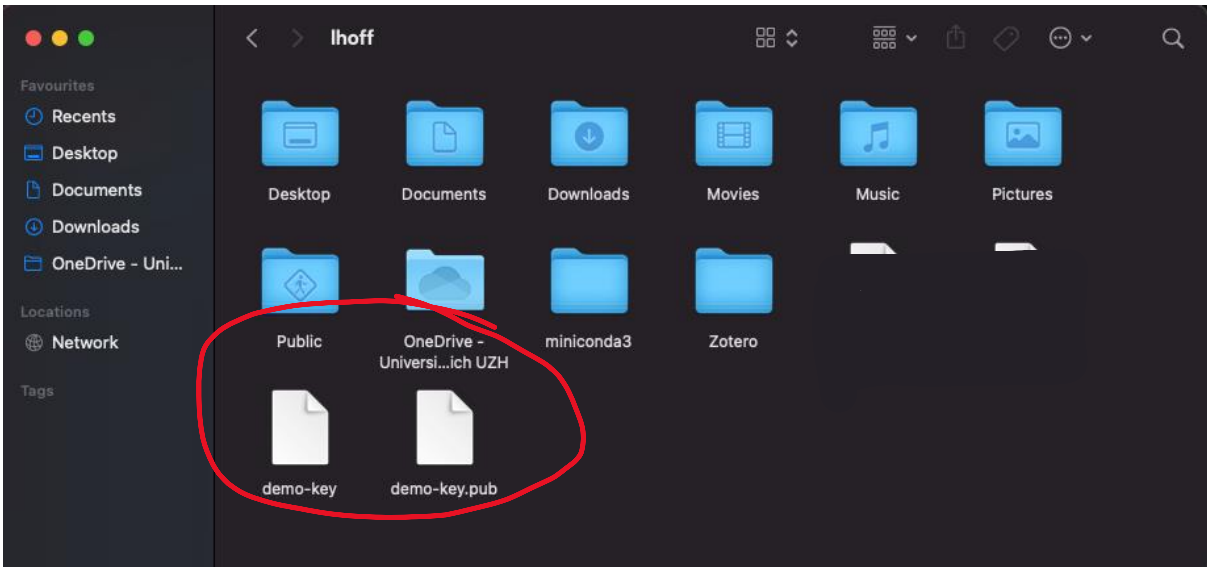
To use Git and transfer data from your local machine to the online repository you will need to prove to GitHub that you are allowed to do so. For this purpose, SSH keys are used. Setting up SSH keys involves the following steps:

- Create a SSH key pair. This can be done via the command line/terminal using the following command: ssh-keygen -t rsa -b 4096 -C email@demo.com. Make sure to replace the dummy email with the email address you used to create you GitHub account. You will then be asked to enter a name for the key, and if you want to secure it with a password. After this the key will be generated and the terminal looks as follows:

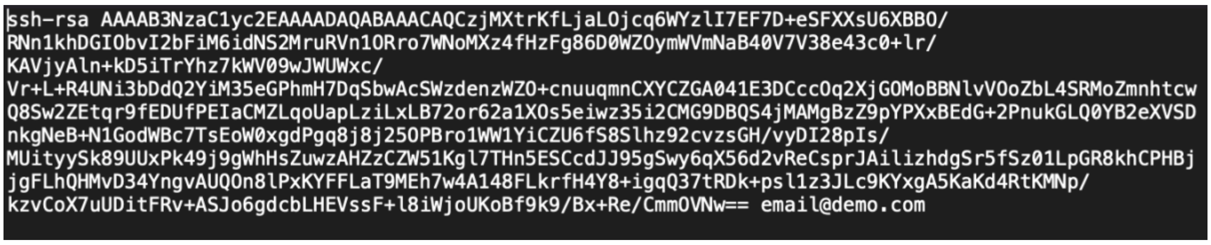


```
[(base) Lukass-iMac:~ lhoff$ ssh-keygen -t rsa -b 4096 -C email@demo.com
Generating public/private rsa key pair.
Enter file in which to save the key (/Users/lhoff/.ssh/id_rsa): demo-key
[Enter passphrase (empty for no passphrase):
[Enter same passphrase again:
Your identification has been saved in demo-key.
Your public key has been saved in demo-key.pub.
The key fingerprint is:
SHA256:yj3F+tPgwzUBCGDxliyEB2ZkXgtbPS111plfbdmrPhc email@demo.com
The key's randomart image is:
+---[RSA 4096]----+
| .BoB+.+..o. o  +|
| =.B.++.oo. +  .=|
|  o.o =o   . . o.|
|     o   .  . .. |
|        S o  ..  |
|     . o o. o. E |
|      o +o +..  .|
|         o= .o . |
|          .o  o  |
+----[SHA256]-----+
```

- If you check your /Users/user directory, you will find the key pair. The .pub key will be uploaded on GitHub, while the other key should remain on your mach



- Open the .pub key with a text editor and copy it.



```
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAACAQCzjMXtrKfLjaL0jcq6WYzlI7EF7D+eSFXXsU6XBBO/
RNn1khDGIObvI2bFiM6idNS2MruRVn1ORro7WNoMXz4fHzFg86D0WZOymWVmNaB40V7V38e43c0+lr/
KAVjyAln+kD5iTrYhz7kWV09wJWUWxc/
Vr+L+R4UNi3bDdQ2YiM35eGPhmH7DqSbwAcSWzdenzWZO+cnuuqmnCXYCZGA041E3DCccOq2XjGOMoBBNlvVOoZbL4SRMoZmnhtcw
Q8Sw2ZEtqr9fEDUfPEIaCMZLqoUapLziLxLB72or62a1XOs5eiwz35i2CMG9DBQS4jMAMgBzZ9pYPXxBEdG+2PnukGLQ0YB2eXVSD
nkgNeB+N1GodWBc7TsEoW0xgdPgq8j8j25OPBro1WW1YiCZU6fS8Slhz92cvzsGH/vyDI28pIs/
MUityySk89UUxPk49j9gWhHsZuwzAHZzCZW51Kgl7THn5ESCcdJJ95gSwy6qX56d2vReCsprJAilizhdgSr5fSz01LpGR8khCPHBj
jgFLhQHMvD34YngvAUQOn8lPxKYFFLaT9MEh7w4A148FLkrfH4Y8+igqQ37tRDk+psl1z3JLc9KYxgA5KaKd4RtKMNp/
kzvCoX7uUDitFRv+ASJo6gdcbLHEVssF+l8iWjoUKoBf9k9/Bx+Re/CmmOVNw== email@demo.com
```

- Then go to GitHub and in the Settings to the “SSH and GPG” keys tab. Here, click on “New SSH key”, give it a title, e.g. University PC Key or anything that lets you know for which machine the key is used, and paste your .pub key in the bottom field. Finally, click on “Add SSH key”.
- As final step, we need to tell the local machine that there is a new SSH key it should use. To do so:
   a. Start the ssh-agent in the background: eval “$(ssh-agent -s)”
   b. Open the ssh-config file: open ~/.ssh/config
   c. Paste the following text at the bottom of you ssh-config file:

      ```
      Host *.github.com
        AddKeysToAgent yes
        UseKeychain yes
        IdentityFile ~/.ssh/your-key-name
      ```

   d. Add the ssh-key to the ssh-agent: ssh-add -K ~/.ssh/ your-key-name

For more information on how to set up the ssh-key see:
https://docs.github.com/en/authentication/connecting-to-github-with-ssh/generating-a-new-ssh-key-and-adding-it-to-the-ssh-agent

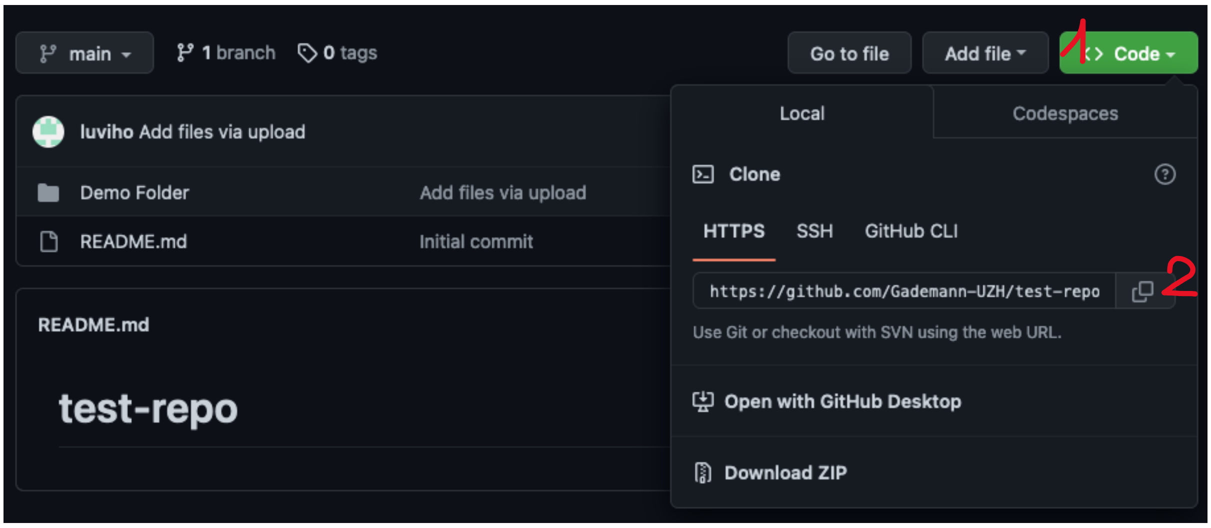
Congrats! Your local machine is now connected to your GitHub account and is ready to be used.

Git – Use

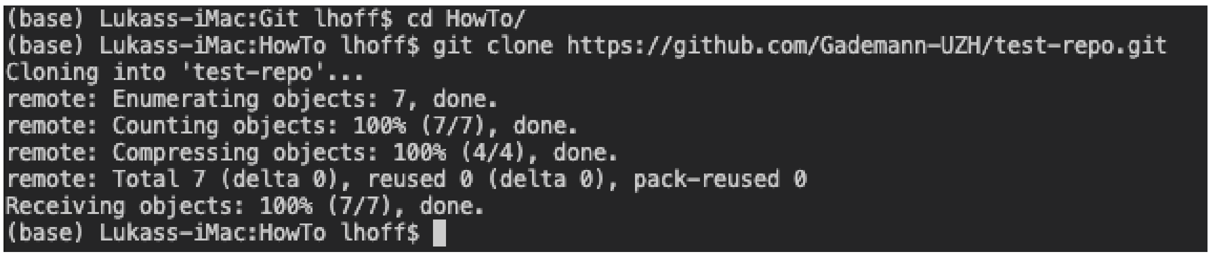
The easiest way to get started with Git and GitHub is to initialize a repository in GitHub as shown above in the “GitHub as Cloud Storage” section. However, instead of uploading the files via the “Add file” button, we can use Git. To do so, open Visual Studio Code with Your Git folder open. If you did not set up your Code editor refer to the respective section above.

In Visual Studio Code open a terminal via the Terminal/New Terminal tab.

To clone the repository to your local machine and automatically connect it, make sure to go to your local git folder by means of the command line. Go to the GitHub repository you want to clone and copy the corresponding link.



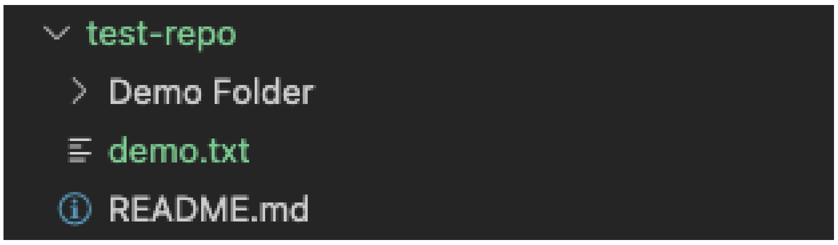
Go back to your code editor and type git clone *paste the copied link* and press enter. If successful your terminal should look similar to this:



```
(base) Lukass-iMac:Git lhoff$ cd HowTo/
(base) Lukass-iMac:HowTo lhoff$ git clone https://github.com/Gademann-UZH/test-repo.git
Cloning into 'test-repo'...
remote: Enumerating objects: 7, done.
remote: Counting objects: 100% (7/7), done.
remote: Compressing objects: 100% (4/4), done.
remote: Total 7 (delta 0), reused 0 (delta 0), pack-reused 0
Receiving objects: 100% (7/7), done.
(base) Lukass-iMac:HowTo lhoff$ █
```

Also you should see a new folder in your local Git folder. All files in this folder are subject to version control with git, which means the program keeps track of changes. In our case, we simply want to use it to upload and maybe update files of our project.

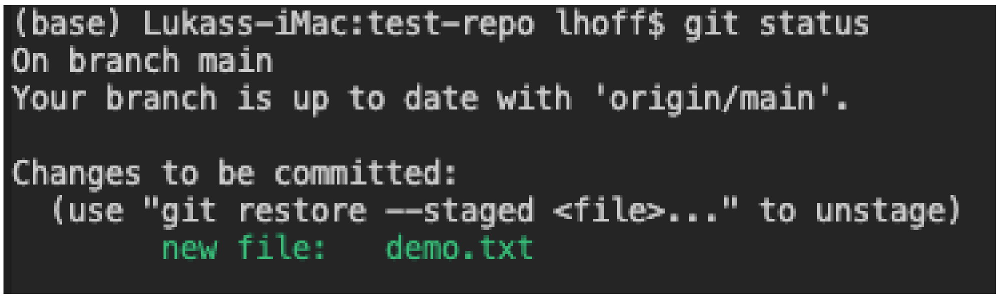
To showcase how this works, we create a new file “demo.txt” and add it to the repository folder. Instead of a .txt file this could also be your project folder, the workflow is the same.



To tell Git to track the file we use the add command: «git add demo.txt». In case weh ave multiple files that we want to track, i.e. a whole project folder, we can use «git add .» instead. This will add all files within the current directory.

With the “git status” command we can check which files have been added or what their status is.

If we run this for the current file it will look similar to this:



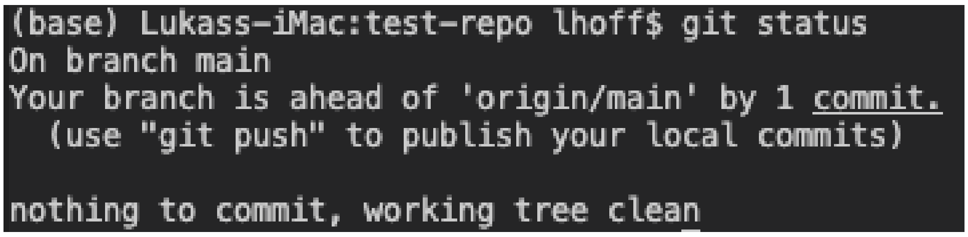
```
(base) Lukass-iMac:test-repo lhoff$ git status
On branch main
Your branch is up to date with 'origin/main'.

Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
        new file:   demo.txt
```

We can see that a new file has been added. In the case of adding a project folder many files would be listed.

To tell git to take a snapshot of the current file state and get everything ready for the upload with use the “git commit” command. This command, however, requires an additional input, which is a message that will be associated with the changes we made. To give this input we add -m “my message” to the command. A fitting command for this example case would be: “git commit -m “Created demo.txt””

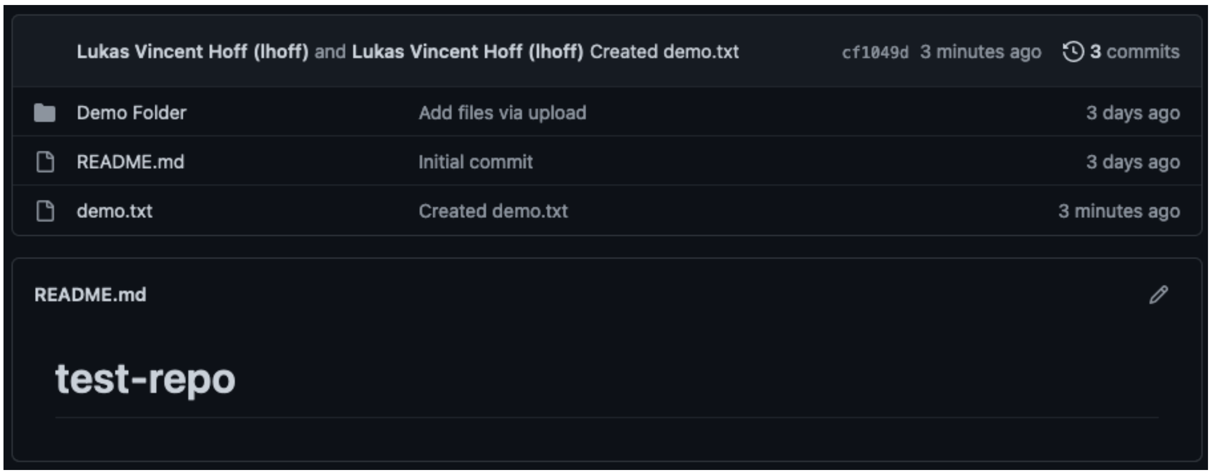
Using “git status” again will now tell us that our version is currently ahead of the online repository, and we are ready to push our files online.



```
(base) Lukass-iMac:test-repo lhoff$ git status
On branch main
Your branch is ahead of 'origin/main' by 1 commit.
  (use "git push" to publish your local commits)

nothing to commit, working tree clean
```

For this use the “git push origin main” command. If you now reload the GitHub page of your repository, you will see the demo.txt file in the list of files.



Lukas Vincent Hoff (lhoff) and Lukas Vincent Hoff (lhoff) Created demo.txt cf1049d 3 minutes ago 3 commits

Demo Folder	Add files via upload	3 days ago
README.md	Initial commit	3 days ago
demo.txt	Created demo.txt	3 minutes ago

README.md

test-repo

Further information:

Git and GitHub for Beginners: https://www.youtube.com/watch?v=RGOj5yH7evk.